

# Dialectical Logic K-Model: the Discrete Time Dynamical Sampling System, Multidimensional Logic Variable and Associate Database(ADB)

Yaozhi Jiang<sup>1</sup>

<sup>1</sup> Shijiazhuang High-tech district, Hebei, China

Correspondence: Yaozhi Jiang, Shijiazhuang High-tech district, Hebei, China. E-mail: jiangyaozhi@126.com

Received: September 11, 2017 Accepted: October 8, 2017 Online Published: March 5, 2018

doi:10.5539/jmr.v10n2p88

URL: <https://doi.org/10.5539/jmr.v10n2p88>

## Abstract

Following the earlier works about dialectical logic K-model by the author, in this succeed paper author described the three problems : the first is that discrete time dynamical sampling system to solve which the true-value function is unknown and need discrete time dynamical sampling system to obtain a series of sampled discrete time true-value function points to predict the continuous true-value function or we need some properties of true value function in the frequency domain, a several formulas for true-value function of single-dimensional logic variable via discrete Fourier transformation are explained; the second is the graph expression and matrices expression for the multidimensional logic variables in dialectical logic K-model. Multidimensional logic variable is important that can be used in multidimensional contradictions and in multiple-person games. In fact, author also described the graph  $G_K^p$  and corresponding matrices of the multidimensional logic variables; the third is the machine oriented database, associated database i.e. ADB, this is a new database for artificial intelligence, in present paper author describes theoretical properties and some features of ADB.

**Keyword:** Discrete Fourier transformation, Multidimensional logic variable, Graph expression of multidimensional logic variable, Matrix expression of multidimensional logic variable, Associated database

**AMS(MSC2010):** 03B99

## 1. Introduction

In order to make the machine have an ability to thinking and reasoning with dialectical logic established by Hegel (Bencivenga, E. 2000; Kosko, M., 1966; Zhong-Kuan Zhao,2008), author has setup a mathematical model, dialectical logic K-model, for machine (Yaozhi Jiang, 2017). In present paper, author has expanded the results explained(Yaozhi Jiang., 2017) to the three problems below: the first is discrete time dynamical sampling system and single-dimensional discrete time Fourier transform to single-dimensional dialectical logic variable; the second is graph expression and matrices expression of multidimensional logic variable, and the third is a series properties and definitions for associated database(ADB).

## 2. The Discrete Time Dynamical Sampling System

Some kinds of dialectical logic K-model problem is in which the true value function is unknown, in which some process are real-time mode so that they are non-repeated-ability, thus its true value functions need the discrete time dynamical sampling system to decide, or even if the true value function is known but we need its properties in frequency domain, therefor we need the discrete time Fourier transform to treat these problems. The system is called as dynamical, because its sampling pulses are not all ready and are sampled one by one, one sampling pules and one treatment by mathematical process. By the time sequence of sampling pulse, can produce a series of mathematical recurrence formulas. The mathematical methods to reproduce the original function are many, one of them is the discrete time dynamical pules sampling function and single-dimensional Fourier transform. In present paper, author only use the DISCRETE TIME FOURIER TRANSFORMATION, i.e. DTFT, to deal with the problem in which the true value function is unknown, such as  $A_i^{\pm n}(\Delta t)$ , explained in the paper of author.(Yaozhi Jiang, 2017)

## 2.1 Basic definitions and Symbols

2.1.1 Review the definitions used in the paper (Yaozhi Jiang, 2017)

2.1.2 Some symbols and definitions of discrete time sampling system

2.1.2.1 denote the imaginary symbol by

$$j = \sqrt{-1}$$

2.1.2.2 unit sampling pulse

$$\delta(t) = \begin{cases} 1, & \text{if } t = 0; \\ 0, & \text{if } t \neq 0. \end{cases}$$

2.1.2.3 if the function  $f(t)$  is continuous at the time  $t_0$ , then the sample function at time  $t_0$  is

$$f^{\uparrow}(t) = f(t_0)\delta(t - t_0)$$

2.1.2.4 truncation function

$$\chi(t) = \begin{cases} 1, & \text{if } t \in [-\frac{\Delta t}{2}, T_0 - \frac{\Delta t}{2}]; \\ 0, & \text{if } t \notin [-\frac{\Delta t}{2}, T_0 - \frac{\Delta t}{2}]. \end{cases}$$

2.1.2.5 sampling waveform

$$F^{\uparrow}(t) = \sum_{i=-\infty}^{+\infty} f(i\Delta t)\delta(t - i\Delta t)$$

2.1.2.6 truncated sampling waveform

$$F_T^{\uparrow}(t) = \left( \sum_{i=-\infty}^{+\infty} f(it)\delta(t - i\Delta t) \right) \chi(t) = \sum_{i=0}^{n-1} f(it)\delta(t - i\Delta t)$$

In the formula as above,  $n = \frac{T_0}{\Delta t}$ .

2.1.2.7 sampling pulse function to the spectral function

$$\Delta^{\uparrow}(t) = T_0 \sum_{r=-\infty}^{+\infty} \delta(t - rT_0)$$

2.1.2.8 discrete convolution  $C(t)$

$$C(t) = F_T^{\uparrow}(t) * \Delta^{\uparrow}(t) = \left( \sum_{i=0}^{n-1} f(i\Delta t)\delta(t - i\Delta t) \right) * \left( T_0 \sum_{r=-\infty}^{+\infty} \delta(t - rT_0) \right)$$

2.1.2.9 Fourier transform  $\Omega(f)$  to the discrete convolution  $C(t)$  and its inverse Fourier transform  $\Omega^{-1}(t)$

$$\Omega(f) = \Omega\left(\frac{s}{n\Delta t}\right) = \sum_{i=0}^{n-1} \Omega^{-1}(i\Delta t) e^{-j\frac{2s\pi i}{n}}, \quad s = 0, 1, 2, \dots, n;$$

$$\Omega^{-1}(t) = \Omega^{-1}(i\Delta t) = \frac{1}{n} \sum_{s=0}^{n-1} \Omega\left(\frac{s}{n\Delta t}\right) e^{j\frac{2s\pi i}{n}}, i = 0, 1, 2, \dots, n-1;$$

in the formula as above,  $f$  is the frequency variable.

2.1.2.10 recurrence Fourier transformation  $\Omega_n(f)$  and  $\Omega_n^{-1}(t)$  while the  $n$  is increasing

If the sampling number is  $n$ , then

$$\Omega_n(f) = \Omega_n\left(\frac{s}{n\Delta t}\right) = \sum_{i=0}^{n-1} \Omega_n^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n}}, s = 0, 1, 2, \dots, n; \Omega_n^{-1}(t) = \Omega_n^{-1}(i\Delta t) = \frac{1}{n} \sum_{s=0}^{n-1} \Omega_n\left(\frac{s}{n\Delta t}\right) e^{j\frac{2\pi si}{n}},$$

$$i = 0, 1, 2, \dots, n-1;$$

If the sampling number is  $(n+1)$ , then

$$\Omega_{n+1}(f) = \Omega\left(\frac{s}{(n+1)\Delta t}\right) = \sum_{i=0}^n \Omega_{n+1}^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n+1}}, s = 0, 1, 2, \dots, (n+1);$$

$$\Omega_{n+1}^{-1}(t) = \Omega_{n+1}^{-1}(i\Delta t) = \frac{1}{n+1} \sum_{s=0}^n \Omega_{n+1}\left(\frac{s}{(n+1)\Delta t}\right) e^{j\frac{2\pi si}{n+1}}, i = 0, 1, 2, \dots, n;$$

2.2 The True-Value Function  $T(A_i^{\pm n}(t))$ ,  $A_i^{\pm n}(x, y, z; \Delta t)$ ,  $A_i^{\pm n}(w; \Delta t)$  in Sampling

**Remark1:** In the condition of synchronizing sampling, the power function matrix  $M_{KP}^{\uparrow}$ , flow function matrix  $M_{KF}^{\uparrow}$ , and the contradiction function matrix  $M_{KC}^{\uparrow}$  explained in the paper are still existence (Yaozhi Jiang, 2017), they are also satisfied by the Kirchhoff law and expressed in pulse-sampling function forms. In present paper, we don't concern with the three matrices.

2.2.1 The recurrence Fourier transform and inverse transform of the true value function  $T(A_i^{\pm n}(t))$  in sampling

For the discrete convolution

$$C(t) = T_T^{\uparrow}(A(t))(t) * \Delta^{\uparrow}(t) = \left( \sum_{i=0}^{n-1} f(i\Delta t) \delta(t - i\Delta t) \right) * \left( T_0 \sum_{r=-\infty}^{+\infty} \delta(t - rT_0) \right)$$

If the sampling number is  $n$ , then

$$\Omega_n(f) = \Omega_n\left(\frac{s}{n\Delta t}\right) = \sum_{i=0}^{n-1} \Omega_n^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n}}, s = 0, 1, 2, \dots, n; \Omega_n^{-1}(t) = \Omega_n^{-1}(i\Delta t) = \frac{1}{n} \sum_{s=0}^{n-1} \Omega_n\left(\frac{s}{n\Delta t}\right) e^{j\frac{2\pi si}{n}},$$

$$i = 0, 1, 2, \dots, n-1;$$

If the sampling number is  $(n+1)$ , then

$$\Omega_{n+1}(f) = \Omega\left(\frac{s}{(n+1)\Delta t}\right) = \sum_{i=0}^n \Omega_{n+1}^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n+1}}, s = 0, 1, 2, \dots, (n+1);$$

$$\Omega_{n+1}^{-1}(t) = \Omega_{n+1}^{-1}(i\Delta t) = \frac{1}{n+1} \sum_{s=0}^n \Omega_{n+1}\left(\frac{s}{(n+1)\Delta t}\right) e^{j\frac{2\pi si}{n+1}}, i = 0, 1, 2, \dots, n;$$

2.2.2 The recurrence Fourier transform and inverse transform of the true value function  $A_i^{\pm n}(x, y, z; \Delta t)$ , when  $x, y, z$  are all constants in sampling

For the discrete convolution

$$C(t) = T^{\uparrow}(A(t)) * \Delta^{\uparrow}(t) = \left( \sum_{i=0}^{n-1} f(i\Delta t) \delta(t - i\Delta t) \right) * \left( T_0 \sum_{r=-\infty}^{+\infty} \delta(t - rT_0) \right)$$

If the sampling number is  $n$ , then

$$\Omega_n(f) = \Omega_n\left(\frac{s}{n\Delta t}\right) = \sum_{i=0}^{n-1} \Omega_n^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n}}, s = 0, 1, 2, \dots, n; \Omega_n^{-1}(t) = \Omega_n^{-1}(i\Delta t) = \frac{1}{n} \sum_{s=0}^{n-1} \Omega_n\left(\frac{s}{n\Delta t}\right) e^{j\frac{2\pi si}{n}},$$

$$i = 0, 1, 2, \dots, n-1;$$

If the sampling number is  $(n+1)$ , then

$$\Omega_{n+1}(f) = \Omega\left(\frac{s}{(n+1)\Delta t}\right) = \sum_{i=0}^n \Omega_{n+1}^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n+1}}, s = 0, 1, 2, \dots, (n+1);$$

$$\Omega_{n+1}^{-1}(t) = \Omega_{n+1}^{-1}(i\Delta t) = \frac{1}{n+1} \sum_{s=0}^n \Omega_{n+1}\left(\frac{s}{(n+1)\Delta t}\right) e^{j\frac{2\pi si}{n+1}}, i = 0, 1, 2, \dots, n;$$

2.2.3 The recurrence Fourier transform and inverse transform of the true value function  $A_i^{\pm n}(w; \Delta t)$ , when  $w$  is a constant in sampling

For the discrete convolution

$$C(t) = T^{\uparrow}(A(t)) * \Delta^{\uparrow}(t) = \left( \sum_{i=0}^{n-1} f(i\Delta t) \delta(t - i\Delta t) \right) * \left( T_0 \sum_{r=-\infty}^{+\infty} \delta(t - rT_0) \right)$$

If the sampling number is  $n$ , then

$$\Omega_n(f) = \Omega_n\left(\frac{s}{n\Delta t}\right) = \sum_{i=0}^{n-1} \Omega_n^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n}}, s = 0, 1, 2, \dots, n; \Omega_n^{-1}(t) = \Omega_n^{-1}(i\Delta t) = \frac{1}{n} \sum_{s=0}^{n-1} \Omega_n\left(\frac{s}{n\Delta t}\right) e^{j\frac{2\pi si}{n}},$$

$$i = 0, 1, 2, \dots, n-1;$$

If the sampling number is  $(n+1)$ , then

$$\Omega_{n+1}(f) = \Omega\left(\frac{s}{(n+1)\Delta t}\right) = \sum_{i=0}^n \Omega_{n+1}^{-1}(i\Delta t) e^{-j\frac{2\pi si}{n+1}}, s = 0, 1, 2, \dots, (n+1);$$

$$\Omega_{n+1}^{-1}(t) = \Omega_{n+1}^{-1}(i\Delta t) = \frac{1}{n+1} \sum_{s=0}^n \Omega_{n+1}\left(\frac{s}{(n+1)\Delta t}\right) e^{j\frac{2\pi si}{n+1}}, i = 0, 1, 2, \dots, n;$$

**Remark2:** At first, the formulas in section 2.2.1., 2.2.2. and 2.2.3. are similar as above, because of in the section 2.2.2., sampling for three-dimensional Euclidean space logic variable  $A_i^{\pm n}(x, y, z; \Delta t)$  is only limited by that  $x, y, z$  are all constants, and in the section 2.2.3., sampling for the logic variable  $A_i^{\pm n}(w; \Delta t)$  acting on topology network space is also only limited by that  $w$  is a constant, of cause these results are not perfect. It is necessary to explain these remainder problems via succeed papers in future.

### 3. Multidimensional Logic Variable

Multidimensional logic variable is used for multiple-contradiction or multiple-person game. These are huge kinds of problems.

Following the paper(Yaozhi Jiang, 2017), denote the  $P$ -dimensional logic variable as

$$W_p^{\pm n}(t) = \{A_1^{\pm n}(t), A_2^{\pm n}(t), \dots, A_p^{\pm n}(t)\}, \quad (1)$$

and its true value function as  $T(W_p^{\pm n}(t))$ ; the corresponding graph expression to  $P$ -dimensional variable  $W_p^{\pm n}(t)$  is as  $G_K^P$ .

#### 3.1 Definitions and Symbols of Multidimensional Logic Variable

Defines the definitions and symbols of multidimensional logic variable as below.

##### 3.1.1 The definition for multidimensional logic variable $W_p^{\pm n}(t)$

As shown in formula (1) above, the complicated relationships among  $P$  of 1-dimension logic variable  $A_p^{\pm n}(t)$  must be considered. There are two kinds of points set, one kind of them is the positive points set

$$v_{1,1}^+(t), v_{1,2}^+(t), \dots, v_{1,n}^+(t); v_{2,1}^+(t), v_{2,2}^+(t), \dots, v_{2,n}^+(t); \dots, v_{p,1}^+(t), v_{p,2}^+(t), \dots, v_{p,n}^+(t)$$

and another kind of them is the negative points set

$$v_{1,1}^-(t), v_{1,2}^-(t), \dots, v_{1,n}^-(t); v_{2,1}^-(t), v_{2,2}^-(t), \dots, v_{2,n}^-(t); \dots, v_{p,1}^-(t), v_{p,2}^-(t), \dots, v_{p,n}^-(t)$$

every 1-dimension logic variable  $A_p^{\pm n}(t)$  all are related to the two kinds of points. Thus the formula (1) above can be

understood a function of multidimensional logic variable satisfied Kirchhoff's law. Almost a best way to define the

multidimensional logic variables is via the graph  $G_K^P$ . To see the Fig.1.(Yaozhi Jiang, 2017), the Fig.1. is for

1-dimensional logic variable, and the Fig.1., Fig.2, in this paper are for 2-dimensional logic variable and 3-dimensional logic variable separately.

One general definition for multidimensional logic variable via the graph  $G_K^P$  is shown as below, to see the "3.1.2. the expression by graph  $G_K^P$  to  $P$ -dimensional logic variable  $A_p^{\pm n}(t)$ ".

##### 3.1.2 The expression by graph $G_K^P$ to $P$ -dimensional logic variable $A_p^{\pm n}(t)$ (Berge, C., 1973)

3.1.2.1 Defines a connected and directed simple graph  $G_K^P$ , which has  $V$  nodes and  $E$  edges, i.e.

$$G_K^P = \{V; E\}$$

Obviously,

$$V = 2p(2p-1)(n-1) + 2p + 2$$

$$E = p(2p-1)(4(n-1)+1) + 2p + 1$$

##### 3.1.2.2 Kirchhoff's power-function law

As same as the definition for 1-dimensional logic variable in the paper(Yaozhi Jiang, 2017), in graph  $G_K^P$  every node will be given a power-function  $v_i(t)$  for node  $V_i$ , and a power-function  $v_j(t)$  for node  $V_j$ . If the edge  $E_{ij} \in E$ , then on the edge  $E_{ij}$  exists a power-function

$$e_{ij}(t) = v_j(t) - v_i(t) \quad (2)$$

In formula (2), tail node is left and head node is right, if in the opposite direction would sign a negative sign in the front of  $e_{ij}(t)$ .

For the every cycle  $W_{ij}$  in the graph is satisfied by

$$\sum_{\text{if } E_{ij} \in W_{ij}} e_{ij}(t) = 0$$

### 3.1.2.3 Kirchhoff's flow-function law

In the graph  $G_K^p$ , every node  $V_i$  is satisfied by

$$\sum_{\text{if } E_i \in V_i} f_i(t) = 0$$

As above incoming flow  $f_t(t)$  is positive sign and outgoing flow  $f_i(t)$  is negative sign.

### 3.1.2.4 Contradiction-function

Define

$$R_{ij}(t) = \frac{e_{ij}(t)}{f_{ij}(t)}$$

is the contradiction-function of the edge  $E_{ij}$  in the  $G_K^p$ .

### 3.1.2.5 Work and energy conservation law

For every edge  $E_{ij}$  in the graph  $G_K^p$ , let

$$D_{ij} = \int_{t_0}^{t_1} f_{ij}(t) e_{ij}(t) dt$$

Then  $D_{ij}$  is called that work done in interval  $[t_0, t_1]$  by contradiction-function  $R_{ij}(t)$  on edge  $E_{ij}$ .

Work and energy conservation law: the total algebraic sum of work done on every edge is equal to the total algebraic sum of energy produced in the graph  $G_K^p$ .

### 3.1.2.6 Structure in graph $G_K^p$ for multidimensional logic variable $A_p^{\pm n}(t)$

#### 3.1.2.6.1 There are

$$V = 2p(2p-1)(n-1) + 2p + 2$$

nodes and

$$E = p(2p-1)(4(n-1)+1) + 2p + 1$$

edges in the graph  $G_K^p$ , in which exists a couple of node, one is the positive source node  $V_{+1}$  and another is negative sink node  $V_{-1}$ .

#### 3.1.2.6.2 Graph $G_K^p$ is a connected, directed, simple graph.

#### 3.1.2.6.3 In the graph $G_K^p$ , exists

$$V^+ = p((2p-1)(n-1)+1)$$

nodes connected to positive source node  $V_{+1}$ , these nodes can be called as positive nodes, and another

$$V^- = p((2p-1)(n-1)+1)$$

nodes connected to negative sink node  $V_{-1}$ , these nodes can be called as negative nodes.

3.1.2.6.4 Source node  $V_{+1}$  and sink node  $V_{-1}$  is also connected by one edge  $E_{+1,-1}$ ; the power function of source node  $V_{+1}$  is a constant +1, and power function of sink node  $V_{-1}$  is a constant -1; the power function of the edge  $E_{+1,-1}$  is

a constant +2, the flow function of the edge  $E_{+1,-1}$  is a constant  $I$ .

### 3.1.2.6.5 Primary sub-graph $G_{KP}^p$ and secondary sub-graph $G_{KS}^p$ .

For 1-dimensional logic variable  $A_1^{\pm n}(t)$ , its primary sub-graph  $G_{KP}^1$  and secondary sub-graph  $G_{KS}^1$  is shown in Fig.1. and Fig.2. below separately, for 2-dimensional logic variable  $A_2^{\pm n}(t)$  and 3-dimensional logic variable  $A_3^{\pm n}(t)$ , their primary sub-graph  $G_{KP}^2$  and  $G_{KP}^3$  are shown in Fig.3. and Fig.4. below separately, their secondary sub-graph is the same as the secondary sub-graph  $G_{KS}^1$  of 1-dimensional logic variable shown in Fig.2. as below.

3.1.2.6.6 In the graph  $G_K^p$ , every nodes all be defined power-function  $v_i(t)$ , and every edges all be defined power-function  $e_{ij}(t)$ , flow-function  $f_{ij}(t)$  and contradiction-function  $R_{ij}(t)$ . These parameters are satisfied the Kirchhoff's law as above.

### 3.2 The Definitions of Power Function Matrix $M_{KE}^p$ , Flow Function Matrix $M_{KF}^p$ and Contradiction Function Matrix $M_{KC}^p$ Corresponding to Graph $G_K^p$

The definitions of power function matrix  $M_{KE}^p$ , flow function matrix  $M_{KF}^p$  and contradiction function matrix  $M_{KC}^p$  corresponding to graph  $G_K^p$  is similar to the definitions shown in last paper(Yaozhi Jiang,2017). Therefore no repeating for them in present paper.

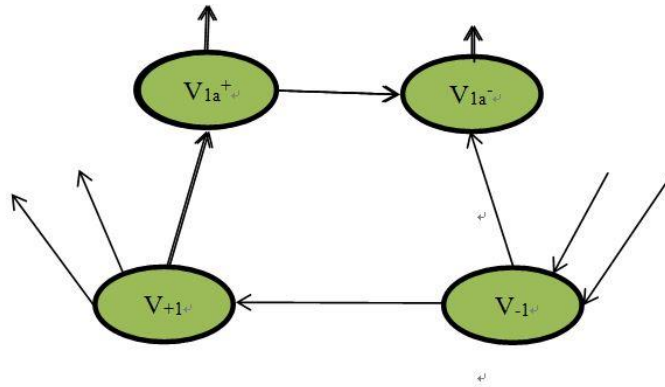


Fig.1. The structure of 1-dimensional primary sub-graph  $G_{KP}^1$

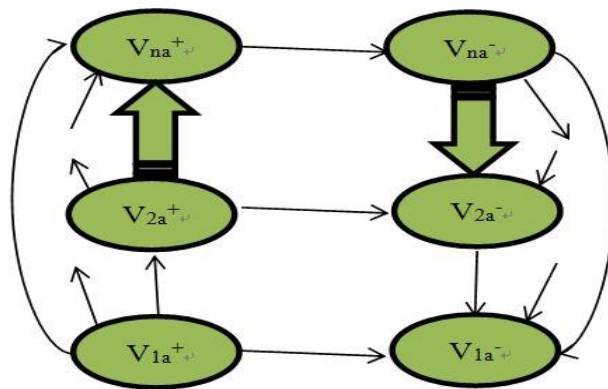
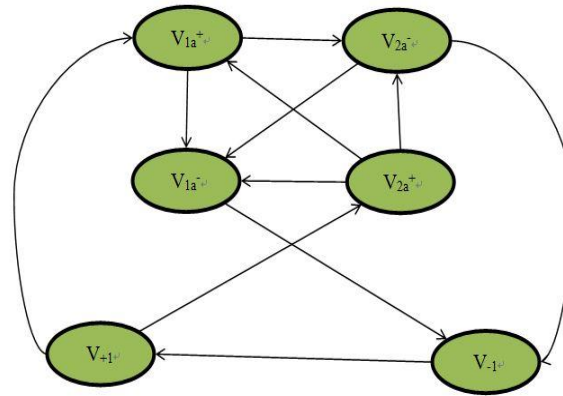
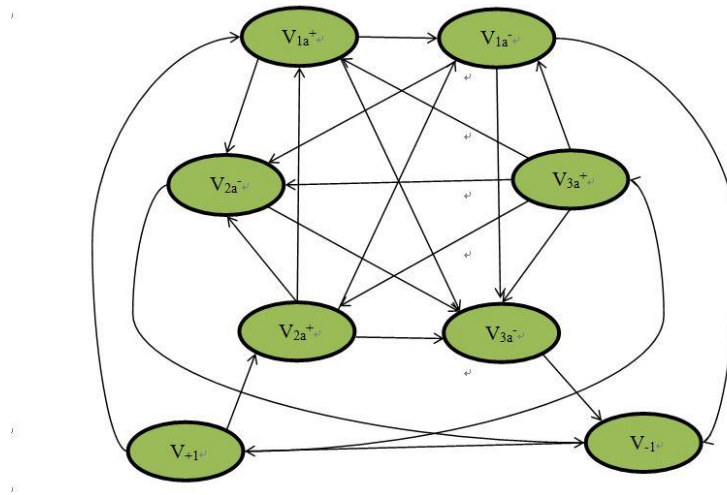


Fig.2. The structure of 1-dimensional secondary sub-graph  $G_{KS}^1$

Fig.3. The structure of 2-dimensional primary sub-graph  $G_{KP}^2$ Fig.4. The structure of 3-dimensional primary sub-graph  $G_{KP}^3$ 

### 3.3 The Definitions for Multidimensional Variable True Value Function $T(W_p^{\pm n}(t))$ and True Value Function Matrix $M_{KT}^p$

Define the true value function for every secondary sub-graph of multidimensional logic variable

$$T(W_p^{+n}(t)) = \frac{1}{n} \sum_{i=1}^n v_i^{+}(t)$$

or

$$T(W_p^{-n}(t)) = \frac{1}{n} \sum_{i=1}^n v_i^{-}(t)$$

So we obtained  $2p$  values for the multidimensional logic variable, its true value function matrix is defined as below:

- At the first row is the power function of  $2p$  nodes of primary sub-graph;
- The first column is transposed vector of the first row;
- The elements of matrix  $M_{KT}^p$  is the corresponding column element minus the corresponding row element;
- The main diagonal elements of matrix  $M_{KT}^p$  all are zero;
- The matrix is negative axial symmetrical on the main diagonal.



The key of the matrix is sequencing the matrix elements, except the elements of first row and first column, from large to small. The larger the number is, the more superior the logic variable corresponded to the number is, the largest number is the best superior.

$$M_{KT}^p =$$

$$\begin{vmatrix} 0 & T(A_1^{+n}(t)) & T(A_2^{+n}(t)) & \cdots & T(A_p^{+n}(t)) & T(A_1^{-n}(t)) & T(A_2^{-n}(t)) & \cdots & T(A_p^{-n}(t)) \\ T(A_1^{+n}(t)) & 0 & T(A_1^{+n}(t)) & \cdots & T(A_1^{+n}(t)) & T(A_1^{+n}(t)) & T(A_1^{+n}(t)) & \cdots & T(A_1^{+n}(t)) \\ T(A_2^{+n}(t)) & T(A_2^{+n}(t)) & -T(A_2^{+n}(t)) & \cdots & -T(A_p^{+n}(t)) - T(A_1^{-n}(t)) - T(A_2^{-n}(t)) & \cdots & -T(A_p^{-n}(t)) \\ T(A_2^{+n}(t)) & -T(A_1^{+n}(t)) & 0 & \vdots & T(A_2^{+n}(t)) & T(A_2^{+n}(t)) & T(A_2^{+n}(t)) & \cdots & T(A_2^{+n}(t)) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ T(A_p^{+n}(t)) & T(A_p^{+n}(t)) & T(A_p^{+n}(t)) & \cdots & 0 & T(A_p^{+n}(t)) & T(A_p^{+n}(t)) & \cdots & T(A_p^{+n}(t)) \\ T(A_1^{-n}(t)) & -T(A_1^{+n}(t)) - T(A_2^{+n}(t)) & \cdots & 0 & -T(A_1^{-n}(t)) - T(A_2^{-n}(t)) & \cdots & -T(A_p^{-n}(t)) \\ T(A_1^{-n}(t)) & T(A_1^{-n}(t)) & T(A_1^{-n}(t)) & \cdots & T(A_1^{-n}(t)) & 0 & T(A_1^{-n}(t)) & \cdots & T(A_1^{-n}(t)) \\ T(A_1^{-n}(t)) & -T(A_1^{+n}(t)) - T(A_2^{+n}(t)) & \cdots & -T(A_p^{+n}(t)) & 0 & -T(A_2^{-n}(t)) & \cdots & -T(A_p^{-n}(t)) \\ T(A_2^{-n}(t)) & T(A_2^{-n}(t)) & T(A_2^{-n}(t)) & \cdots & T(A_2^{-n}(t)) & T(A_2^{-n}(t)) & 0 & \cdots & T(A_2^{-n}(t)) \\ T(A_2^{-n}(t)) & -T(A_1^{+n}(t)) - T(A_2^{+n}(t)) & \cdots & -T(A_p^{+n}(t)) - T(A_1^{-n}(t)) & 0 & \cdots & -T(A_p^{-n}(t)) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ T(A_p^{-n}(t)) & T(A_p^{-n}(t)) & T(A_p^{-n}(t)) & \cdots & T(A_p^{-n}(t)) & T(A_p^{-n}(t)) & T(A_p^{-n}(t)) & \cdots & 0 \\ T(A_p^{-n}(t)) & -T(A_1^{+n}(t)) - T(A_2^{+n}(t)) & \cdots & -T(A_p^{+n}(t)) - T(A_1^{-n}(t)) - T(A_2^{-n}(t)) & \cdots & 0 \end{vmatrix}$$

These matrices shown above would be compute-able in machine, therefor the multidimensional dialectical logic variables would be compute-able in machine.

#### 4. The Theoretical Outlines of Associated Database (ADB)

Traditional database is user oriented database, in the paper author will describe a machine oriented database: associated database (ADB).

##### 4.1 Information and Unprocessed Information

As shown in author's last paper (Yaozhi Jiang, 2017), the object consists of five factors which are MATTER, ENERGY, INFORMATION, SPACE and TIME. Therefor the information can be divided into two types: information inner machine and information outer machine. The information outer machine can be transformed into information inner machine only by the three-step: Sensation, Abstraction and Thinking. Without sensation machine is blind like a person without sensory organ such as ears, eyes, nose, etc.

For machine information can be divided into two kinds: the one is objective information, and another is subjective information.

Subjective information also can be divided into two kinds: one of them is unprocessed information, i.e. the information only sensed into machine by sensor and didn't be processed by the other two steps: abstraction and thinking. As a opposite thing, another kind of them is all-processed information.

For the information, of cause the machine have to memory the information that the machine interest in. Therefor the need of database is produced, the require of machine memory to the subjective information is powerful driving for database.

##### 4.2 Sensation, Abstraction and Thinking of Machine

Sensation is input-ability to machine from objective information with sensors. Abstraction is classification-ability for sensed information properties and conceptualize it. Thinking is ability researching from concepts to get their property-relations and subjective logical relations.

Sensation, abstraction and thinking are all divided into direct or indirect, i.e. direct-sensation and indirect-sensation;

direct-abstraction and indirect-abstraction; direct-thinking and indirect-thinking. All of direct-types are getting from researching and all of indirect-types are getting from database stored formerly or from the communication inter-machine.

Previously the research operator  $\nabla$  and inverse operator  $\nabla^{-1}$ , will be explained in succeed paper in future, are important in the process of sensation, abstraction and thinking.

#### 4.3 Memory by Machine

The three main recognition technologies (image recognition technology, voice recognition technology and text recognition technology) and other recognition technologies will produce a huge of subjective information need to memory. This is the original requirement for database.

A database system generally consists of database and database managing system. At present, we only interest in the theory of database and do not discuss the physical structure of database.

The three main recognition technologies (image recognition technology, voice recognition technology and text recognition technology) and other recognition technologies will produce three databases (image database, voice database and text database) and other databases, such as physical knowledge database, chemical knowledge database, mathematical knowledge database, etc.

As shown above, what information need to store and apply to thinking and reasoning in machine will produce database.

#### 4.4 The Operations to Data in Associated Database(ABD)

Because of ABD is a machine oriented database, we can define a series of operations in ADB. These operations are stored into and data output, relationship constructing and relationship reconstructing, compression and decompression, save and delete, remark, sort, insert, copy, inquiry, etc. Such operations are all powered by the database managing system.

#### 4.5 Data Model Features for Machine Oriented Database

As shown below, see the Fig.5., data model of ABD is explained as a graph mode. In the graph some definitions must be discussed as below.

Tricycle Venn diagram only are drawn on the graph, actually must be drawn  $m$ -cycle Venn diagram in the graph and every cycle is denoted separately by image database, voice database, text database, physical knowledge database, chemical database, mathematical knowledge database, and etc. The common sets of these  $m$ -cycle Venn diagram are perfectly.

The “P-relation” is denoted to “Properties relationship”, “C-relation” to “Causality relationship”,

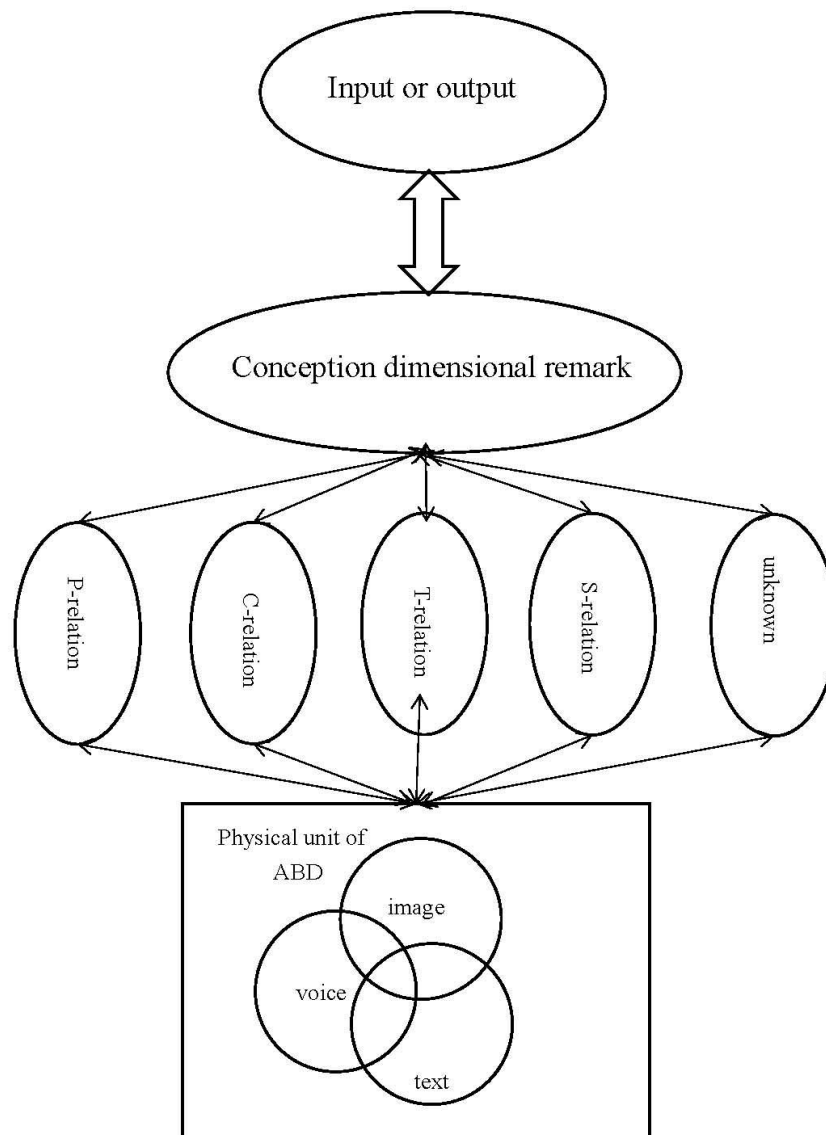


Fig.5. The structure graph of associate database

“T-relation” to “Time-sequence relationship”, “S-relation” to “Space-sequence relationship”, and “unknown” to “relationship unknown”.

These relationships all appear the feature that “one to many”, “one to one”, “many to one” and “many to many”.

Within ABD there are two sub-levels: the parent is named event-file and the child is named data-record.

Conception-dimensional number is the limitation number of the conception. For example, A RUNNING WHITE HOUSE is a 4-dimensional conception, and the conception is defined on the four dimensions: number-dimension, action-dimension, colour-dimension and animal-dimension. In the ABD, the conception-dimensional relationship is the properties of conceptions and events.

Sensation, abstraction and thinking is three-step in artificial intelligence. Actually text is compression from image, voice and other information by sensors.

When we say “this is a hill”, in fact we just do the second-step, i.e. abstraction, only obtain a conception-classification, but not a truth. With another word, the second-step is solution of “what is this?”, however only the third-step, i.e. thinking, is just the solution of “why ( or who or where or when) can cause this?”.

Now the most difficult in artificial intelligence is just the third-step, i.e. thinking, but the difficult limited by the second-step, i.e. abstraction, is solving one by one.

#### 4.6 The communication Inter-Machine

The artificial intelligence have to need a special programming language for machine, but till now the programming language is not coming. This difficult maybe overcome at the future not far.

The communication inter-machine is a important role in artificial intelligence. The communication inter-machine is what information transforms inter-machine. At first, we can determine the properties information transformation as below:

- a. Information can be divided into is objective information and subjective information.
- b. Information is no disappearance in information transforming, i.e. suppose that machine A has information I and has given the information I to machine B, then machine A still has the information I.
- c. In transformation process, information is easy to distortion, therefore some errors must be produced in transforming process.

The carrier of transformed information is a variety of physical factors and chemical factors, such as sonic wave, light wave, electromagnetic wave, mechanic force, chemical parameters, etc.

What all of factors of matter, energy, space and time are possible carriers loaded information.

By the way, human language may be a role in communication inter-machine.

### 3. Conclusion

Now we have obtained some formulas about the situation of discrete time dynamical sampling system via discrete Fourier transformation in section 2; the graph expression and matrices expression of the multidimensional logic variable involved the 3-dimensional Euclidean space, even though author treated the three space dimension as constants, in section 3, therefore the multidimensional logic variable is compute-able in machine; and some results of associated database (ADB) which is a database oriented by machine never described by any author. This paper is a succeed and developing paper of the last paper (Yaozhi Jiang, 2017), and author will publish a series of other new papers to research the article in dialectical logic K-model.

### References

- Berge, C. (1973). Graph and Hyper graphs [M]-Amsterdam: North-Holland Publishing Company.
- Bencivenga, E. (2000). Hegel's Dialectical Logic, Oxford University Press.
- Kosko, M. (1966). the Formalization of Hegel' Dialectical logic, *International Philosophical Quarterly*, IPQ64: 596-631, January. <https://doi.org/10.5840/jpq.19666456>
- Yaozhi Jiang. (2017). Dialectical logic K-Model: A Mathematical Model for Machine, *Journal of Mathematics Research*, 9(6). <https://doi.org/10.5539/jmr.v9n6p82>
- Zong-kuan Zhao. (2008). From Classical logic to Mathematical Dialectical Logic, *Journal of Bijie University*, 2008-01(in Chinese).

### Note:

Some parts of this paper has published in the journal : Open Access Library Journal 4, 2017(ISSN Print: 2333-9705, ISSN Online: 2333-9721) entitled "Axiom System and Theorems for Dialectical Logic K-Model", DOI:10.4236/oalib.1103817. Now this paper has been filled by some new materials from researches and re-entitled to publish.

### Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).